SOLUTION DETECTION DDoS ATTACK IN SOFTWARE DEFINED NETWORK

Tran Thi Nga*, Bui Thu Giang
Academy of Cryptography Techniques

ARTICLE INFO

Received: 03/7/2023
Revised: 20/9//2023
Published: 20/9//2023

KEYWORDS

Software-defined Network (SDN)
DDoS Attack
Mininet
Opendaylight
Snort

ABSTRACT

Software-defined network (SDN) is a technology trend that big giants like Google, Facebook and Amazon apply and deploy into data centres to realize centralized data management and easy-to-scale network capabilities. However, this technology also has many potential risks and challenges, leading to loss of data and disrupting users' ability to provide services. One of the most significant challenges is Distributed Denial of Service (DDoS) attacks on SDN due to their intense impacts on the system's availability and reliability. Many state-of-the-art studies have been published to detect DDoS attacks, but these methods are only focused on the control layer of SDN. This paper aimed to detect a DDoS attack in the data layer of SDN by adopting hping3 and snort tools performed on a mininet simulator. The experimental results demonstrated that a DDoS attack was successfully initialized and then detected by snort in order to warn the administrator about the network connection status.

GIẢI PHÁP PHÁT HIỆN TẤN CÔNG DDoS TRONG MẠNG ĐỊNH NGHĨA MỀM

Trần Thị Ngà*, Bùi Thu Giang
Học viện Kỹ thuật mật mã

THÔNG TIN BÀI BÁO

Ngày nhận bài: 03/7/2023
Ngày hoàn thiện: 20/9//2023
Ngày đăng: 20/9//2023

TỪ KHÓA

Mạng định nghĩa mềm (SDN)
Tấn công DDoS
Mininet
Opendaylight
Snort

TÓM TẮT

Mạng định nghĩa mềm (SDN) đang là xu hướng công nghệ mà các ông lớn như Google, Facebook và Amazon ứng dụng và triển khai trong các trung tâm dữ liệu để thực hiện quản lý dữ liệu tập trung và dễ dàng mở rộng mạng. Tuy nhiên, công nghệ này cũng tiềm ẩn nhiều rủi ro và thách thức dẫn đến việc thất thoát dữ liệu và làm gián đoạn khả năng truyền tải cung cấp các dịch vụ cho người dùng. Một trong những thách thức lớn nhất là các cuộc tấn công từ chối dịch vụ phân tán (DDoS) vì đây là những cuộc tấn công có ảnh hưởng nghiêm trọng đến tính sẵn sàng và tính tin cậy của hệ thống. Nhiều nghiên cứu gần đây đã được đưa ra để phát hiện tấn công DDoS, tuy nhiên các phương pháp này đều chỉ tập trung ở lớp điều khiển của SDN. Bài báo này với mục tiêu là phát hiện tấn công DDoS ở lớp dữ liệu của SDN bằng cách sử dụng công cụ hping3 và snort được mô phỏng trên mininet. Kết quả thử nghiệm đã chứng minh các cuộc tấn công DDoS đã được khởi tạo thành công và sau đó được phát hiện bởi snort để cảnh báo tới người quản trị về tình trạng kết nối mạng.

DOI: https://doi.org/10.34238/tnu-jst.8251

* Corresponding author. *Email: tranthinga@actvn.edu.vn*

1. Giới thiệu

Trên thế giới hiện có nhiều định nghĩa về mạng định nghĩa mềm (SDN) nhưng theo tổ chức ONF thì SDN là một kiểu kiến trúc mạng mới, năng động, dễ quản lý, chi phí hiệu quả, dễ thích nghi và rất phù hợp với nhu cầu mạng ngày càng tăng hiện nay. Kiến trúc này phân tách chức năng điều khiển mạng và chức năng vận chuyển dữ liệu, điều này cho phép việc điều khiển mạng có thể lập trình được dễ dàng và hạ tầng mạng vật lý độc lập với các ứng dụng và dịch vụ mạng [1].

Đối với SDN, việc điều khiển được tập trung tại lớp điều khiển, các thiết bị mạng chỉ có nhiệm vụ chuyển tiếp gói tin. Điều này dẫn đến nhiều thách thức an ninh, do đó, nhu cầu bảo vệ mạng khỏi các cuộc tấn công khác nhau trở nên vô cùng quan trọng. Tấn công từ chối dịch vụ phân tán (DDoS) trong SDN là cuộc tấn công có ảnh hưởng tới toàn bộ các lớp của SDN, tấn công này khai thác băng thông và giới hạn khả năng mở rộng cơ sở hạ tầng của SDN [2]. Trước khi thực hiện giảm thiểu tấn công, thì bước đầu tiên là phải phát hiện được tấn công trên SDN. Nhiều nhà nghiên cứu đã có các công bố liên quan đến việc phát hiện tấn công DDoS vào SDN [3] – [6]. Trong [3], các tác giả đề xuất đến phương pháp phát hiện tấn công DDoS sử dụng tập lệnh python hoạt động tự động để phát hiện tấn công DDoS với sự hỗ trợ của Openflow. Tuy nhiên, phương pháp này đòi hỏi người quản trị mạng phải có hiểu biết sâu sắc về ngôn ngữ python cũng như về openflow của SDN. Trong [4], tác giả đã đề xuất một phương pháp phát hiện và ngăn chặn DDoS sớm bằng snort. Họ sử dụng bộ điều khiển Ryu SDN cho thử nghiệm của mình nhưng các tác giả bị giới hạn ở các tham số được sử dụng, các công cụ để kiểm tra, số lượng gói bị tràn ngập trong một loại tấn công DDoS là ICMP flood. Gần đây, công cụ phát hiện DDoS dựa trên entropy khác đã được đề xuất, nhưng họ chỉ giới hạn ở kiểu tấn công DDoS UDP flood [5], [6]. Do đó, từ các công trình hiện có liên quan đến công cụ phát hiện DDoS, có thể thấy rằng các công cụ hoặc hệ thống có sẵn đều bị hạn chế. Hơn nữa, các phương pháp phát hiện tấn công đều diễn ra ở lớp điều khiển của SDN mà không quan tâm đến tấn công và phát hiện tấn công DDoS ở các lớp khác của SDN.

Xuất phát từ thực tế trên, bài báo này sẽ đề xuất giải pháp phát hiện tấn công DDoS ở lớp dữ liệu của SDN bằng cách sử dụng công cụ hping 3 và snort [7]. Theo hiểu biết của tác giả, đây là công cụ đã được sử dụng nhiều trong mạng truyền thống nên các nhà quản trị mạng khi vận hành và sử dụng công cụ này có thể dễ dàng cài đặt, cấu hình mà vẫn đảm bảo được việc phát hiện tấn công DDoS trong SDN [8]. Thực nghiệm của bài báo được mô phỏng trên mininet với việc sử dụng bộ điều khiển opendaylight [9]. Ngoài ra, bài báo cũng tiến hành thực hiện nhiều kiểu tấn công DDoS với các loại khác nhau như: ICMP flood, TCP SYN Flood mà các bài báo trước đó chỉ thực hiện một trong hai loại.

Bài viết được bố cục theo 4 mục chính: Sau phần 1 giới thiệu, phần 2 trình bày phương pháp nghiên cứu. Phần 3 tiến hành thực hiện mô phỏng trên cơ sở đó đưa ra kết quả và thảo luận. Cuối cùng là kết luận và hướng phát triển.

2. Phương pháp nghiên cứu

2.1. Kiến trúc của mạng SDN

Kiến trúc của SDN bao gồm: Lớp ứng dụng (Application Layer), lớp điều khiển (Control Layer) và lớp cơ sở hạ tầng/lớp dữ liệu (Infrastructure Layer/ data layer). Các phần sẽ liên kết với nhau thông qua giao thức hoặc các API. Hình 1 mô tả kiến trúc của SDN một cách đơn giản và đầy đủ [1], [2].

Lớp ứng dụng (Applycation Layer)

Là các ứng dụng được triển khai trên mạng, kết nối tới lớp điều khiển thông qua các API, cung cấp khả năng cho phép ứng dụng lập trình lại (cấu hình lại) mạng (điều chỉnh các tham số trễ, băng thông, định tuyến…) thông qua lớp điều khiển lập trình giúp cho hệ thống mạng hoạt động tối ưu theo một yêu cầu nhất định.

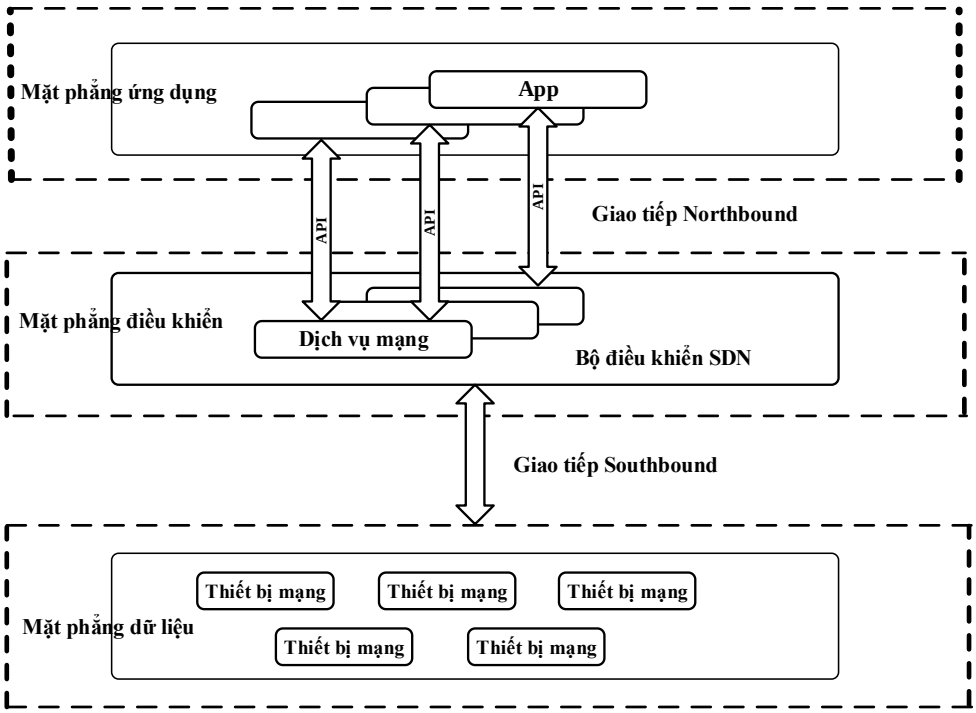
Lớp điều khiển (Control Layer)

Là nơi tập trung các bộ điều khiển (controller) thực hiện việc điều khiển cấu hình mạng theo các yêu cầu từ lớp ứng dụng và khả năng của mạng. Các bộ điều khiển này có thể là các phần mềm được lập trình.

Bộ điều khiển là một ứng dụng quản lý kiểm soát luồng lưu lượng trong môi trường mạng. Để truyền thông điều khiển lớp cơ sở hạ tầng, lớp điều khiển sử dụng các cơ chế như Openflow, ONOS, ForCES, PCEP, NETCONF, SNMP hoặc thông qua các cơ chế riêng biệt. Hầu hết các bộ điều khiển của SDN hiện nay dựa trên giao thức Openflow.

Lớp cơ sở hạ tầng/dữ liệu (Infrastructure Layer/Data layer)

Lớp dữ liệu (cơ sở hạ tầng) của hệ thống mạng, bao gồm các thiết bị mạng thực tế (vật lý hay ảo hóa) thực hiện việc chuyển tiếp gói tin theo sự điều khiển của lớp điều khiển. Một thiết bị mạng có thể hoạt động theo sự điều khiển của nhiều bộ điều khiển khác nhau, điều này giúp tăng cường khả năng ảo hóa của mạng.



Hình 1. *Kiến trúc của SDN*

2.2. Tấn công DDoS vào mạng SDN

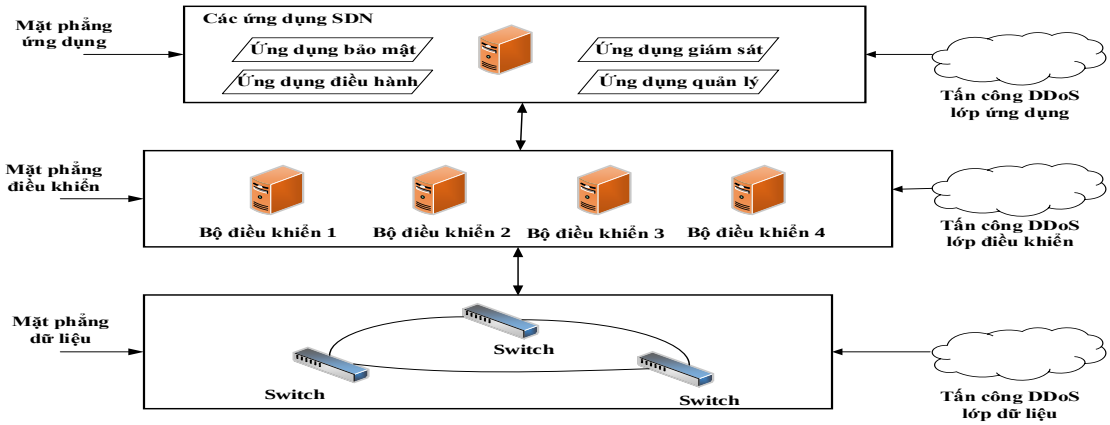
Các cuộc tấn công DDoS tiềm năng có thể được thực hiện trong cả 3 mặt phẳng của SDN như mô tả hình 2.

Tấn công DDoS vào lớp ứng dụng: Có hai cách khác nhau để triển khai các cuộc tấn công DDoS vào lớp ứng dụng. Một là các cuộc tấn công tập trung vào một số ứng dụng của SDN; Hai là tấn công tập trung vào giao tiếp giữa bộ điều khiển với phần mềm quản lý (Northbound API) của SDN [2]. Các cuộc tấn công DDoS ở lớp ứng dụng được thiết kế để tấn công các ứng dụng cụ thể, phổ biến nhất là máy chủ web hoặc các dịch vụ như thoại SIP và BGP. Tấn công DDoS vào lớp ứng dụng có thể ảnh hưởng đến ứng dụng khác không phải là mục tiêu. Các cuộc tấn công DDoS như vậy thường có cường độ từ thấp đến trung bình vì chúng phải tuân theo giao thức mà ứng dụng đang sử dụng, thường liên quan đến bắt tay giao thức và tuân thủ giao thức/ứng dụng. Điều này có nghĩa là các cuộc tấn công DDoS này chủ yếu sẽ được thực hiện bằng cách sử dụng các máy khách thông minh riêng, thường là các thiết bị IoT và không thể bị

giả mạo. Các ứng dụng bị xâm nhập như vậy gây ra vấn đề lớn đối với mạng SDN. Có thể kể đến các loại tấn công như: Slow Read, Slowloriss, Slowpost…

Tấn công DDoS vào lớp điều khiển: Lớp điều khiển là trung tâm của SDN và có thể coi là lớp dễ bị tấn công nhất vì nó kết nối toàn bộ an ninh trong SDN. Các cuộc tấn công vào lớp này cố gắng làm cho các người dùng hợp pháp không thể truy cập được các chức năng của bộ điều khiển bằng cách làm cạn kiệt tài nguyên tính toán hoặc bộ nhớ. Kẻ tấn công có thể tạo ra một luồng lưu lượng mạng khổng lồ trong thời gian ngắn bằng cách sử dụng máy chủ lưu trữ của chính chúng hoặc bằng cách kiểm soát các máy chủ zombie phân tán khác [3]-[6].

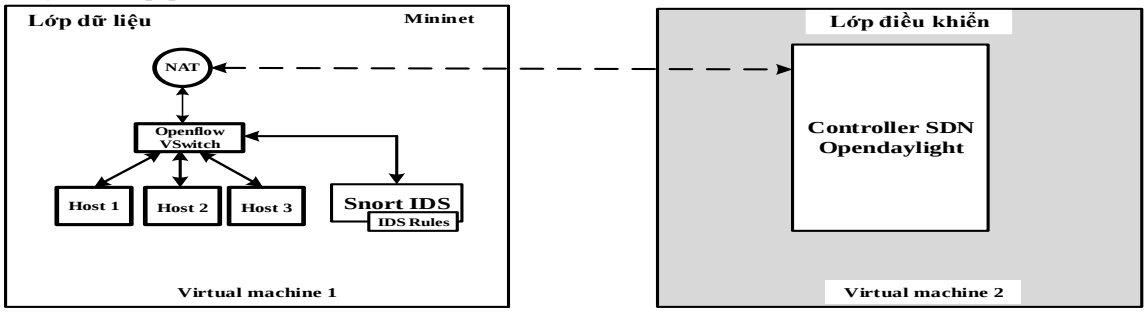
Tấn công DDoS vào lớp dữ liệu: Mặt phẳng dữ liệu không cho phép bất kỳ ứng dụng xấu nào có thể cài đặt, thay đổi hoặc điều chỉnh các quy tắc luồng, khi tấn công DDoS được khởi động thì các bộ định tuyến bị tấn công. Trong cuộc tấn công DDoS, một số lượng lớn các gói tin được gửi đến các nhóm máy chủ trong mạng. Phần lớn switch (bộ chuyển mạch) không thể phát hiện ra tấn công, và gói tin được gửi đến bộ điều khiển. Cả gói tin hợp pháp và gói tin giả mạo DDoS kết hợp với nhau có thể làm cạn kiệt tài nguyên của bộ điều khiển. Như vậy, tấn công DDoS vào lớp dữ liệu cũng gián tiếp làm bộ điều khiển bị ảnh hưởng [2].



Hình 2. *Tấn công DDoS vào mạng SDN*

2.3. Giải pháp phát hiện tấn công DDoS ở lớp dữ liệu

Trong các giải pháp phát hiện tấn công DDoS trước và gần đây đều tập trung tấn công DDoS và phát hiện tấn công DDoS ở lớp điều khiển của SDN [3] - [6]. Với các giải pháp phát hiện trước đây hầu như đều dựa vào đặc điểm của giao thức OpenFlow trong SDN, để thực hiện phát hiện tấn công hoặc sử dụng thuật toán phát hiện tấn công DDoS. Điều này, dẫn đến khó khăn trong việc quản lý và vận hành hệ thống SDN. Gần đây, cũng có giải pháp sử dụng snort để phát hiện tấn công DDoS nhưng trong bài báo này snort lại được tích hợp ở lớp điều khiển của mạng SDN để cảnh báo gói tin ICMP flooding. Hơn nữa, trong bài báo này sử dụng bộ điều khiển RYU để tạo SDN [4].



Hình 3. *Mô hình giải pháp phát hiện sử dụng snort*

Bài báo này nhóm tác giả trình bày ý tưởng tấn công DDoS ở lớp dữ liệu của SDN với thành phần tham gia sử dụng bộ điều khiển Opendaylight [9]. Đặc biệt, để thực hiện phát hiện tấn công DDoS ở lớp dữ liệu, nhóm tác giả tích hợp snort ở mặt phẳng dữ liệu của SDN hình 3. Điều này, giúp snort có thể giám sát và đưa ra các cảnh báo về các gói tin TCP SYN flooding và ICMP flooding nhờ các luật đã thiết lập trong snort.

3. Kết quả và thảo luận

3.1. Mô phỏng thực nghiệm

3.1.1. Mô hình thực nghiệm

Để xây dựng mô hình tấn công DDoS vào mạng SDN, nhóm tác giả sử dụng các công cụ sau:

Mininet: Mininet là một công cụ giả lập mạng, bao gồm tập hợp các hosts đầu cuối, các switches, routers và các liên kết trên một Linux kernel. Mininet sử dụng công nghệ ảo hóa (ở mức đơn giản) để tạo nên hệ thống mạng hoàn chỉnh, Mininet cho phép tạo topo mạng nhanh chóng, tùy chỉnh được topo mạng, chạy được các phần mềm thực sự như web servers, TCP monitoring, Wireshark, snort; tùy chỉnh được việc chuyển tiếp gói tin [10].

OpenDaylight: Là phần mềm mã nguồn mở dành cho SDN sử dụng giao thức mở cung cấp khả năng kiểm soát tập trung, có khả năng lập trình được và theo dõi các thiết bị mạng. Giống như nhiều bộ điều khiển SDN khác, OpenDaylight hỗ trợ OpenFlow và cung cấp các giải pháp mạng khác sẵn sàng để cài đặt khi có yêu cầu.

MobaXterm: Là phần mềm giúp giao tiếp Windows với Linux, hỗ trợ truy cập từ xa quản trị các thiết bị mạng như Switch, Router…

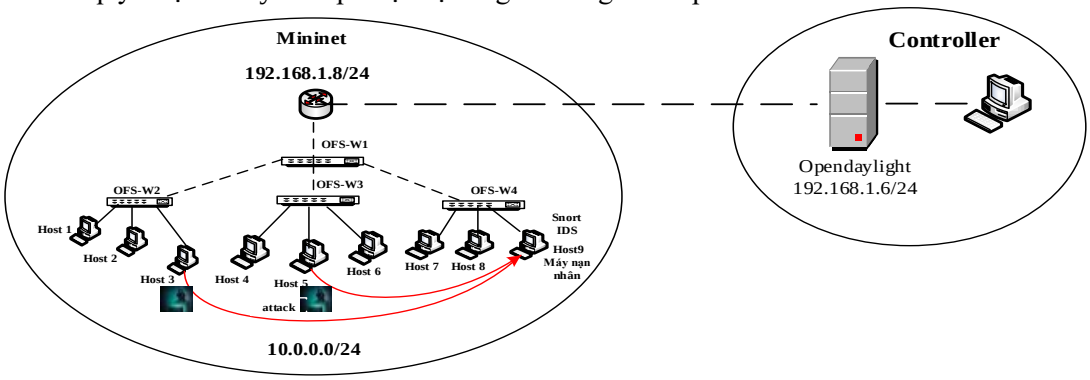
Wireshark: Để chặn bắt gói tin.

Hping3: Công cụ này có thể cấu hình để thực hiện thăm dò mạng, theo dõi, gửi ping, tấn công ngập lụt ở các cấp độ khác nhau... Tóm lại, nó cung cấp cho chúng ta nhiều khả năng để tạo các gói tin và gửi chúng đến các host mục tiêu hoặc các trang web khác nhau. Sử dụng công cụ Hping3 với kỹ thuật tấn công Ping (ICMP) flood, TCP SYN flood để tấn công DDoS vào SDN [11].

Snort: Sử dụng snort để theo dõi lưu lượng mạng nhằm phát hiện ra hiện tượng bất thường, các hoạt động trái phép xâm nhập vào hệ thống SDN.

Mô phỏng tấn công DDoS thực hiện trên máy tính win 10 (64 bit), core i5, RAM 8G. Mô phỏng mạng SDN bằng mã nguồn mở Mininet 2.3.0. Các thành phần của Mininet gồm: các host, các bộ chuyển mạch, bộ điều khiển được ảo hóa. Giả lập này đi kèm với mã nguồn mở có thể tạo một mạng ảo thực tế trên phần cứng thực.

Mô hình thực nghiệm SDN gồm: 1 controller, 4 switch và 9 host. Mỗi switch kết nối trực tiếp với 3 host, các switch này kết nối với một bộ điều khiển C0 qua switch 1 để bộ điều khiển có thể đưa ra các quyết định chuyển tiếp được định nghĩa trong mỗi Open Flow switch hình 4.



Hình 4. *Mô hình thực nghiệm*

Sau khi tạo ra topo mạng theo hình 4, giả định rằng: mô hình chưa có biện pháp nào để ngăn

chặn máy C&C từ ngoài vào, mô hình tạo ra trong số đó có host tạo ra lưu lượng truy cập bình thường, cũng có host bị kẻ tấn công xâm nhập (botnet) để mô phỏng các cuộc tấn công DDoS. Cụ thể: Giả định host 3 và host 5 là host mà kẻ tấn công đã chiếm quyền điều khiển gọi là các bonnet, host 9 là mục tiêu cho các cuộc tấn công và các host còn lại tạo ra lưu lượng truy cập bình thường. Trên host 9 đo lường khả năng truy cập của các host khác trong mạng và hiệu suất sử dụng dịch vụ trước và sau khi bị tấn công.

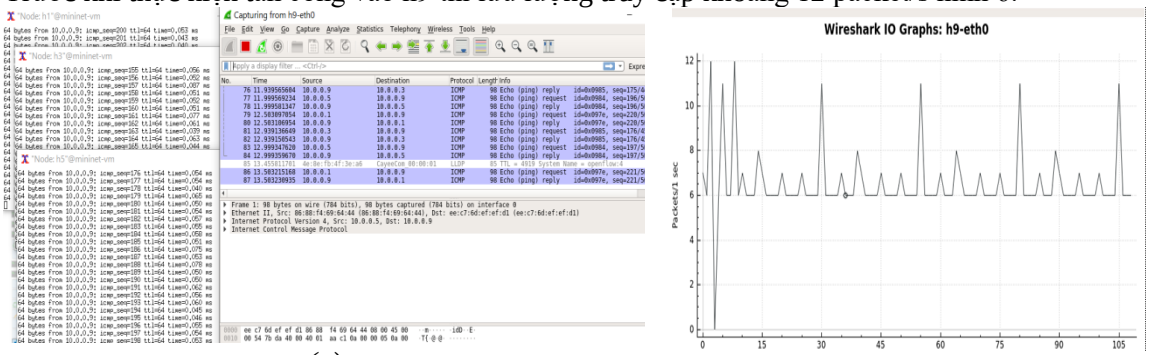
3.1.2. Thực hiện tấn công DDoS

Đầu tiên, cần xây dựng mô phỏng một mạng sử dụng phần mềm mininet với câu lệnh CLI để tạo mạng với 1 controller, 4 switch, 9 host hình 5.

```
mininet@mininet-vm:~$ sudo mn --controller=remote,ip=192.168.1.6,port=6653 --custom topo.py --topo tp
*** Creating network
*** Adding controller
Unable to contact the remote controller at 192.168.1.6:6653
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(s1, s2) (s1, s3) (s1, s4) (s2, h1) (s2, h2) (s2, h3) (s3, h4) (s3, h5) (s3, h6) (s4, h7) (s4, h8) (s4, h9)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9
h2 -> h1 h3 h4 h5 h6 h7 h8 h9
h3 -> h1 h2 h4 h5 h6 h7 h8 h9
h4 -> h1 h2 h3 h5 h6 h7 h8 h9
h5 -> h1 h2 h3 h4 h6 h7 h8 h9
h6 -> h1 h2 h3 h4 h5 h7 h8 h9
h7 -> h1 h2 h3 h4 h5 h6 h8 h9
h8 -> h1 h2 h3 h4 h5 h6 h7 h9
h9 -> h1 h2 h3 h4 h5 h6 h7 h8
*** Results: 0% dropped (72/72 received)
mininet>
```

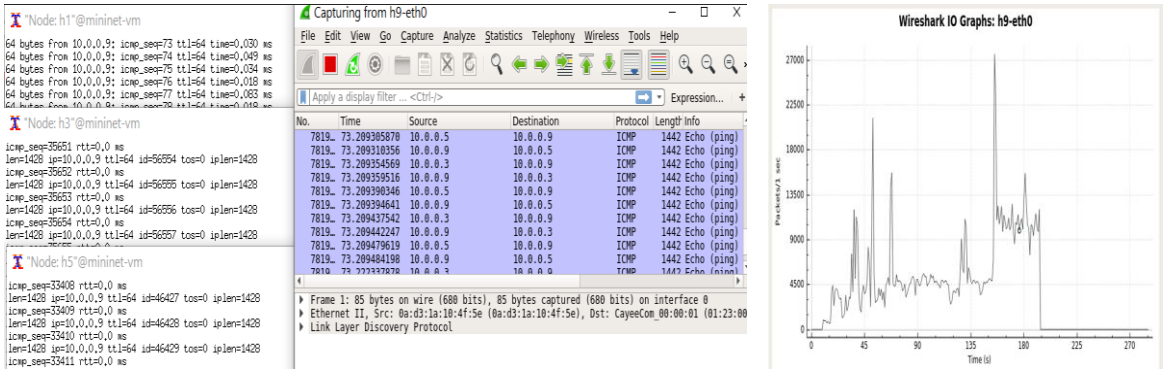
Hình 5. *Tạo mạng SDN*

Sử dụng câu lệnh "*xterm*" để mở terminals của các host: h1, h3, h5 và h9. Hai host h3 và h5 để mô phỏng luồng tấn công DDoS và h1 để mô phỏng luồng lưu lượng truy cập bình thường. Trước khi thực hiện tấn công vào h9 thì lưu lượng truy cập khoảng 12 packet/s hình 6.



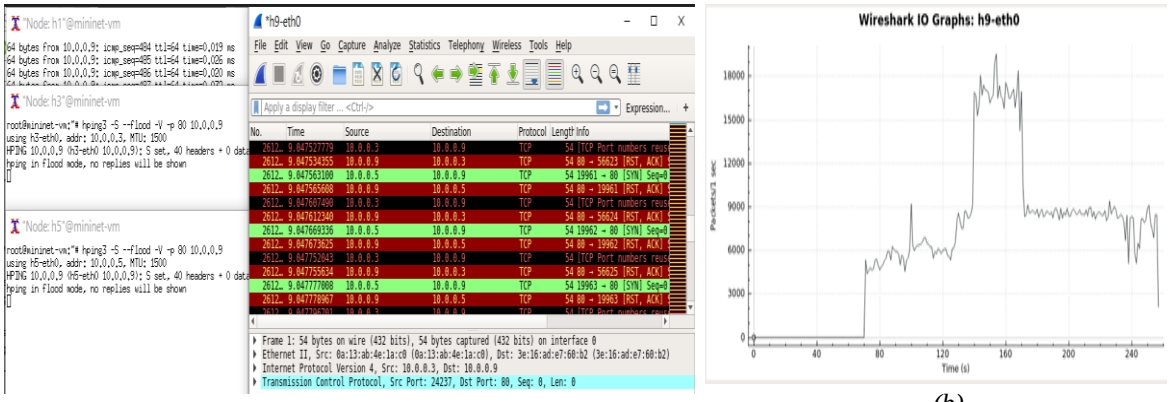
(a) *(b)*

Hình 6. *Khi chưa thực hiện tấn công*



(a) *(b)*

Hình 7. *Tấn công ICMP flooding trên host 9*



(a) *(b)*

Hình 8. *Tấn công TCP SYN flooding trên host 9*

Sử dụng công cụ hping3 gửi gói tin ICMP flood và TCP SYN flood từ host 3 và host 5 tới host 9 trong khi đó host 1 vẫn ping đến host 9, kết quả chỉ ra hình 7 và hình 8 cho thấy số lượng các gói tin lần lượt là 27.000 packets/s và 18.000 packets/s.

3.1.3. Giải pháp phát hiện tấn công DDoS sử dụng snort

Để thực hiện phát hiện tấn công DDoS nhóm tác giả đã tích hợp snort vào mặt phẳng dữ liệu của SDN, cụ thể là host 9 hình 4. Đây là một điểm mới so với các công bố trước đây chỉ tập trung vào lớp điều khiển của SDN [3]-[6], mà không đề cập đến việc phát hiện tấn công ở các lớp khác của mạng SDN. Trong [4] các tác giả cũng có đưa ra giải pháp sử dụng snort tích hợp vào bộ điều khiển RYU kết hợp với giao thức openflow để phát hiện tấn công nhưng giải pháp của họ mới chỉ dừng ở tấn công ICMP Fooding. Đặc biệt, với các giải pháp này đòi hỏi người quản trị mạng phải có những hiểu biết sâu về mạng SDN hoặc phải có những am hiểu về các thuật toán phát hiện tấn công để có thể tích hợp nhằm đưa ra cảnh báo tới người quản trị. Vậy để khắc phục những tồn đọng của các giải pháp trên. Bài báo của chúng tôi sẽ tận dụng ưu điểm của snort là công cụ đã được sử dụng phổ biến và hiệu quả trong các mạng truyền thống để tích hợp vào mặt phẳng dữ liệu của mạng SDN để phát hiện các gói tin không chỉ ICMP flooding mà cả các gói tin TCP flooding mà các bài báo trước đó mới chỉ thực hiện một trong hai loại. Hơn nữa, với giải pháp này giúp các nhà quản trị mạng SDN có cái nhìn đa chiều về cách thức phát hiện tấn công DDoS ở lớp dữ liệu của mạng SDN.

Khi snort được cài đặt thành công, thực hiện cấu hình snort để nó có thể phát hiện những tấn công DDoS bằng cách cấu hình cảnh báo DDoS SDN trong thư mục */etc/snort/*.

```
alert icmp any any -> $HOME_NET any (msg:"ICMP connection attempt"; sid:1000002; rev:1;)
alert tcp any any -> $HOME_NET 80 (flags: S; msg:"Possible DDoS Attack Type : SYNflood";
flow:stateless; sid:3; detection_filter:track by_dst, count 20, seconds 10;)
```

Hình 9. *Các luật tạo trong snort*

Sau khi tạo ra các luật trong file *snort.conf* hình 9, nhóm tác giả khởi tạo một số tấn công DDoS như hình 7 và hình 8. Khi đó, snort chỉ ra cảnh báo cho người quản trị như trong hình 10 và hình 11.

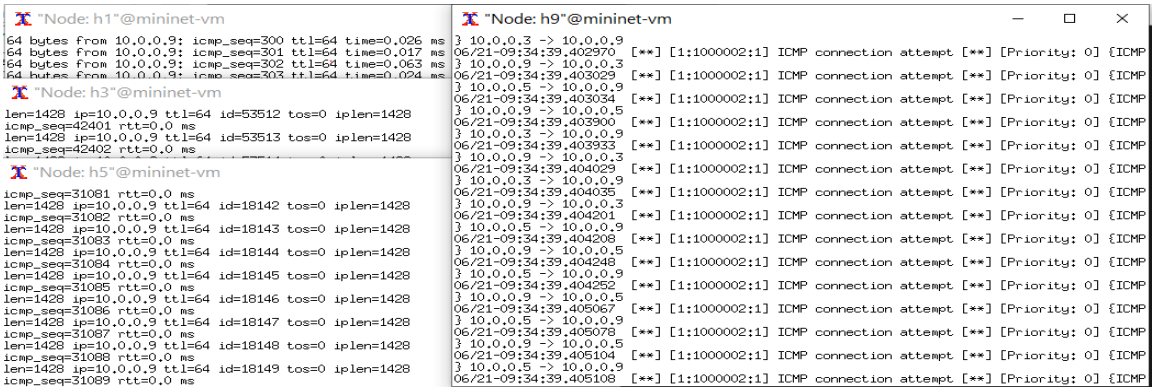
3.2. Thảo luận

Khi chưa thực hiện tấn công DDoS vào mạng SDN, sử dụng công cụ wireshark để chặn bắt gói tin trên host 9 thấy rằng gói tin ICMP từ host 1, host 3, host 5 gửi tới host 9, host 9 vẫn nhận và trả lời kết nối tới các host bình thường hình 6a. Quan sát trên hình 6b thấy rằng số lượng gói tin khi chưa thực hiện tấn công nhiều nhất cũng chỉ khoảng 12packet/s.

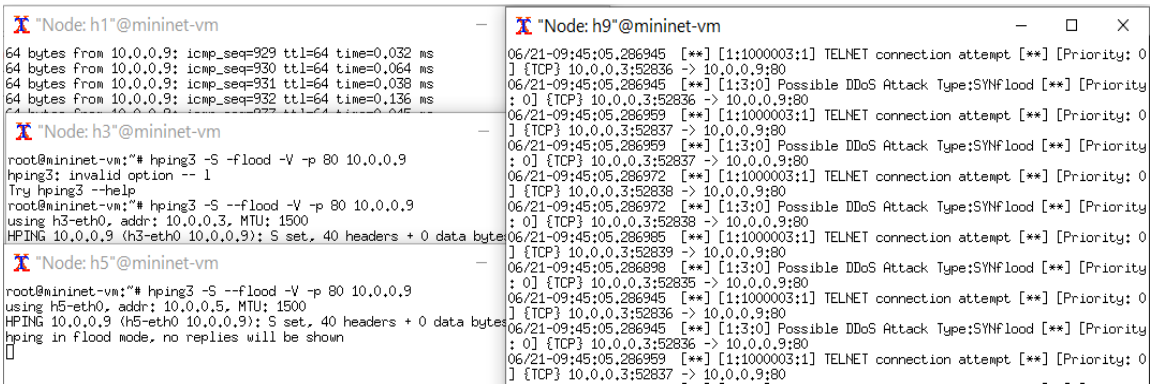
Khi thực hiện tấn công, sử dụng công cụ hping3 với hai trường hợp tấn công khác nhau:

Trường hợp 1: gửi các gói tin ICMP flood từ host 3 và host 5 tới host 9. Đồng thời, lúc này host 1 vẫn đang thực hiện ping đến host 9, nhưng kết quả hình 7a cho thấy host 9 không còn nhận được tín hiệu ping từ host 1. Quan sát trên hình 7b cũng nhận thấy số lượng gói tin lúc này đỉnh điểm lên tới 27000 packet/s; *Trường hợp 2*: sử dụng công cụ hping3 để gửi các gói tin TCP SYN flood từ host 3 và host 5 tới host 9, cùng thời điểm này host 1 vẫn đang thực hiện ping đến host 9 thì kết quả hình 8a sử dụng wireshark cho thấy host 9 không còn nhận được tín hiệu ping từ host 1. Quan sát trên hình 8b có thể thấy số lượng gói tin đỉnh điểm có những lúc lên hơn 18000 packet/s. Như vậy, trong hai trường hợp trên thì có thể nhận thấy host 9 đã bị tấn công dẫn tới không thể nhận và phản hồi các thông tin đến từ host 1.

Trong phần giải pháp: bài báo thực hiện triển khai cài đặt và cấu hình snort ở lớp dữ liệu của SDN. Với các luật đã được thiết lập trong hình 9, giúp người quản trị có thể phát hiện tấn công DDoS với các kiểu ICMP flood và TCP SYN flood hình 10, hình 11. Từ đó, giúp người quản trị mạng có những xử lý kịp thời nhằm giảm thiểu những ảnh hưởng của tấn công DDoS đến SDN.



Hình 10. *Snort cảnh báo tấn công ICMP food trên host 9*



Hình 11. *Snort cảnh báo tấn công SYN food trên host 9*

4. Kết luận và hướng phát triển

Bài báo trình bày tấn công DDoS và giải pháp phát hiện tấn công DDoS sử dụng snort được tích hợp ở lớp dữ liệu trong mạng định nghĩa mềm. Đây là một hướng đi khác so với các nghiên cứu trước đây là chỉ tập trung vào lớp điều khiển. Với đóng góp của bài báo, giúp các nhà quản trị mạng định nghĩa mềm có cái nhìn đa chiều về cách thức thực hiện tấn công DDoS và giải pháp phát hiện tấn công DDoS trong mạng định nghĩa mềm. Hơn nữa, theo hiểu biết của nhóm tác giả thì đây là một giải pháp đơn giản và hiệu quả giúp các nhà quản trị mạng có thể dễ dàng phát hiện tấn công DDoS trong mạng định nghĩa mềm. Hướng nghiên cứu tiếp theo, nhóm tác giả sẽ tập trung vào việc ngăn chặn tấn công DDoS trong SDN.

TÀI LIỆU THAM KHẢO/ REFERENCES

[1] W. Braun and M. Menth, "Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices," *Future Internet*, vol. 6, no. 2, pp. 302–336, 2014.

[2] S. Dong, K. Abbas, and R. Jain, "A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments," *IEEE Access*, vol. 7, pp. 80813-80828, 2019.

[3] H. Abdulkarem and A. Alethawy, "DDoS attack detection and mitigation at SDN environment," *Iraqi Journal of Information and Communications Technology (IJICT)*, vol. 4, no. 1, pp. 1-9, 2021.

[4] P. Manso, J. Moura, and C. Serrão, "SDN-based intrusion detection system for early detection and mitigation of DDoS attacks," *Information*, vol. 10, no. 3, pp. 1-17, 2019.

[5] A. Ahalawat, S. S. Dash, A. Panda, and K. S. Babu, "Entropy Based DDoS Detection and Mitigation in OpenFlow Enabled SDN," *International Conference on Vision towards Emerging Trends in Communication and Networking (ViTECoN)*, 2019, pp.1-5.

[6] M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," *International Conference on Computing, Networking and Communications (ICNC)*, 2015, pp. 77–81.

[7] M.Roesch, "Snort-lightweight intrusion detection for networks," *Proceedings of the 13th USENIX conference on System Administration (LISA '99)*, 1999, pp. 229–238.

[8] A. Gupta and L. S. Sharma, "Performance evaluation of snort and suricata intrusion detection systems on ubuntu server," *in Proceedings of ICRIC*, Springer, 2020, pp. 811–821.

[9] S. Badotra and S. N. Panda, "Evaluation and comparison of OpenDayLight and open networking operating system in software-defined networking," *Cluster Computing*, vol. 23, no. 2, pp. 1281-1291, 2020.

[10] R. L. S. de Oliveira, C. M. Schweitzer, A. A. Shinoda, and L. R. Prete, "Using Mininet for emulation and prototyping Software-Defined Networks," *IEEE Colombian Conference on Communications and Computing (COLCOM)*, 2014, pp. 1-6.

[11] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, "Network attacks: taxonomy, tools and systems," *Journal of Network and Computer Applications*, vol. 40, pp. 307–324, 2014.